Corpus-driven Annotation Enrichment

Felix Kuhr
University of Lübeck
Institute of Information Systems
Ratzeburgerallee 160, 23562 Lübeck
kuhr@ifis.uni-luebeck.de

Bjarne Witten
University of Lübeck
Institute of Information Systems
Ratzeburgerallee 160, 23562 Luebeck
witten@ifis.uni-luebeck.de

Ralf Möller
University of Lübeck
Institute of Information Systems
Ratzeburgerallee 160, 23562 Luebeck
moeller@ifis.uni-luebeck.de

***Abstract*—A reference library can be described as a corpus of an individual composition of documents containing related work of research, documents of favorite authors, or proceedings of a conference. Enriching documents with meaningful annotations is beneficial for the performance of applications like semantic search, content aggregation, automated relationship discovery, query answering and information retrieval. Available (semi-) automatic annotation tools ignore the individual composition of documents in corpora by annotating documents with generic named-entity related data. In this paper, we present and unsupervised corpus-driven annotation enrichment approach considering the composition of documents and use an EM-like algorithm to enrich weakly annotated documents with meaningful annotations of related documents from the same corpus.**

I. Introduction

In linguistics annotations add additional data to documents, supporting humans, and machines to understand the semantic meaning of words in the document. The degree to which *added value* is brought to a document by enriching the document with annotations depends on the benefit for applications like semantic search, aggregation of content, automated relationships discovery, Query-Answering (QA), Information Retrieval (IR), document retrieval (DR), and Knowledge Management (KM).

In recent years, systems have emerged using methods of Information Extraction (IE) [2] and statistical relational learning (SRL) [13] to extract data from the text of million of randomly selected unstructured documents and derive large graph databases (DBs), representing a symbolic content description using entities and relations. Some of the most known systems are DeepDive [23], NELL [10], YAGO [14], FRED [8], and KnowledgeVault [5]. Annotating documents with data from available graph DBs relates to the entity-linking problem that is a well studied field [4], where entities from documents are linked to entities of graphs. However, matching words in the text of documents to entities that are in a graph DB is difficult having no named-entities in the documents. Even if the documents contain named-entities and it is possible to match them to entities in graph DBs, simply annotating documents with entity-related data from graph DBs leads to annotations weakly describing the document's content and ignore the composition of documents. Obviously, collecting documents is not an end in itself and the documents in a corpus might represent related work of research, documents of favorite authors, or selective proceedings of conferences. A subset of annotations of a document's annotation database (ADB) may add value to another document's ADB within the same corpus e.g., by increasing the performance in document retrieval. In this paper, we present an approach to enrich sparse and weakly annotated documents with annotations of documents in the same corpus taking advantage of the higher purpose in mind of people individually selecting the documents in a corpus. We introduce two holistic similarity measures identifying related documents within a corpus and present an unsupervised EM-like algorithm to identify symbolic content descriptions for document. The algorithm has the following properties: (i) Identifying for each document a set of related documents using both, D- and G-similarity. The D-similarity estimates the similarity using the text of documents and the G-similarity works at the annotation-level. (ii) Iteratively enriching ADBs of documents with annotations of relating documents' ADBs, representing a symbolic content description of the documents. (iii) Annotating new, unseen documents using related documents' annotations, and vice versa.

II. Related Work & Preliminaries

Over the recent years, a considerable number of automatic annotation systems have been introduced in the natural language processing (NLP) community. Automatic annotation systems use human language to directly extract data from the text of documents. A well-established technique is named-entity recognition (NER), which is a subtask of IE taking an unannotated block of text and producing an annotated block of text that highlights the names of entities and classify them into predefined categories such as persons, organizations, locations, etc. Generally, automatic annotation systems extract named-entities from the text and use available DBs to identify more entities having a relationship to the extracted entities by using link prediction [9], which is the discipline of estimating the likelihood of the existence of a link between nodes, using the given links and attributes of nodes within a graph [16]. Some well known annotation systems are MINTE [1], Tipalo [7], OpenCalais [12], and BOEMIE [11].

Compared with existing automatic annotation systems, the contributions of this paper are: 1. a novel corpus-driven annotation enrichment approach that considers the composition of documents in the annotation process instead of simply adding data from knowledge bases (KBs) or formulas from ontologies; 2. a flexible annotation approach that allows the annotation

of documents both with additional external data and without external data using well annotated documents to enrich the annotations of other documents within the same corpus.

Topic modeling techniques estimate topics from a collection of documents and calculate for each of the documents a topic probability distribution θ. Topics represent co-occurring words of the documents. The statistical technique called latent Dirichlet allocation (LDA)[17] generates a topic model from a set of documents to identify latent structures such as the topic distribution of documents and word topic distribution. LDA uses a bag of words approach simplifying documents. For document d, LDA learns a discrete probability distribution θ_d that contains for each topic $k \in \{1, \ldots, K\}$ a value between 0 and 1. The sum over all K topics for d is 1. To find topically similar documents we use the Hellinger distance [22] measuring the distance between two probability distributions. Given two topic distributions θ_{d_i} and θ_{d_j} for documents d_i and d_j, the Hellinger distance $H(\theta_{d_i}, \theta_{d_j})$ is given by $\frac{1}{\sqrt{2}}\sqrt{\sum_{k=1}^{K}\left(\sqrt{\theta_{d_i,k}} - \sqrt{\theta_{d_j,k}}\right)^2}$ where k refers to the topics in the documents. Topic modelling techniques reduce the dimensionality of each document to the number of topics k. Having the topic model for the documents within corpus $\mathcal{D}$, it is feasible to calculate the Hellinger distance between documents d_i and d_j. The result is a value between 0 and 1 and $H(\theta_{d_i}, \theta_{d_i}) = 0$. LDA has input documents d_i, $i \in \{1, ..., D\}$, where each document d_i contains words w_n; $n \in \{1, ..., N\}$. The per-word topic assignment $z_{d,n}$ is drawn from a per-document topic distribution vector θ_d. Each topic $k \in \{1, ..., K\}$ is a multinomial distribution of words w. LDA contains two hyperparameters α and β, where α conditions the per-document topic distributions θ_d and β conditions the per-corpus topic distributions ϕ_k, $k \in \{1, ..., K\}$.

Information extraction is a subdomain of NLP referring to methods that extract entities and their relations from text documents. Two main tasks of IE systems are NER and relation extraction. A possible result of an IE system is a set of Resource Description Framework (RDF) triples containing the extractable relations between entities. Identifying entities and relations within arbitrary long sentences containing subordinate clauses and other grammatical structures make IE difficult. Systems are OpenIE [2], Textrunner [15], Gate [6], and the framework document spanners [3]. We use OpenIE which learns a classifier to split sentences of text documents into shorter utterances and apply natural logic [19] to further shorten the utterances in a way such that the shortened utterances can be mapped to OpenIE triples representing subject, predicate, and object.

III. Corpus-driven Annotation Enrichment

In this section, we present the annotation enrichment process enriching ADBs of documents with annotations of related documents in the same corpus. Simply considering all annotations from d-related documents may include many annotations not describing the content of d. One approach to avoid enriching an ADB with annotations that do not describe the content of the corresponding document is to enrich the ADB only with annotations sharing named-entities occurring in the text of the document. Then, new annotations add relations to already known named-entities. However, focusing only on annotations including named-entities not necessarily leads to annotations describing the content of documents. We assume that the annotations of one document add value to another document in the same corpus, if the content of both documents is somehow related. Thus, enriching ADB g_e with annotations from d_e-related documents requires the identification of d_e-related documents and those annotations that are semantically related to the annotations of document d_e. We introduce the D- and G-similarity identifying the d_e-related documents in corpus $\mathcal{D}$ and present an iterative algorithm using the two similarity measures identifying d_e-related documents to assign each annotation with an Expected Relevance Value (ERV) to identify the annotations describing the content of d_e without focusing on named-entities.

A. D-Similarity

D-similarity is based on the idea of topic models and compares the relatedness between two documents using the similarity of the documents' topics. The document-specific topic vector is known as the topic distribution of a document. D-Similarity is defined by: $Sim_D(d_e, d_k) = 1 - H(\theta_{d_e}, \theta_{d_k})$, where $H(\theta_{d_e}, \theta_{d_k})$ estimates the Hellinger distance between the topic distributions of d_e and d_k and $Sim_D(d_e, d_k) \in [0, 1]$. The interval follows directly from the definition of the Hellinger distance. The higher the D-similarity the more similar the documents' topic distribution. The text of documents d_e and d_k having a high D-similarity contain similar content such that annotations for d_k might be added value for d_e.

Comparing two documents using the D-similarity requires both documents having a topic distribution. But, how to compare the D-similarity between a document $d \in \mathcal{D}$ and a new document $d' \notin \mathcal{D}$? Using the parameters of the topic model generated from all documents in $\mathcal{D}$ it is possible to infer the topic distribution for a new document $d' \notin \mathcal{D}$ by applying the *folding in* Gibbs sampling technique [18], which is the same as Gibbs sampling [20], except the sampling bases on the topic-word distribution ϕ and per document-topic distributions θ of the topic model of documents in $\mathcal{D}$. First, for each word w in d' the most probable topic is initialized using ϕ. If d' contains a new word w not part of any document $d \in \mathcal{D}$, we randomly assign the topic. Second, Gibbs sampling estimates the topic distribution of d'. This means that it is only required to perform Gibbs sampling for the words in the new document to infer the topic distribution of document d'. After extending $\mathcal{D}$ with some documents, it is useful to create a new topic model from all documents in $\mathcal{D}$ such that the topic models' parameters depend on all documents.

B. G-Similarity

G-similarity identifies d_e-related documents in $\mathcal{D}$ comparing annotations of g_e with annotations of other documents'

ADB. Each document corresponds to a specific graph ADB which contains the annotations of the corresponding document. Technically, we assume an annotation to be a triple containing a subject (s), predicate (p), and object (o). Comparing the annotations in g_e with those in g_k is the same as identifying subgraph matches between g_e and g_k using labeled vertices and edges in both graphs. Thus, we introduce the G-similarity which identifies subgraph matches between the annotations of two ADBs and bases on the assumption that semantically related documents have at least parts of a subset of annotations in common. We define a similarity function $s(g_e^i, g_k^j)$ calculating a similarity score between two annotations, comparing the i-th annotation in g_e with the j-th annotation in g_k using the entities and relations to estimate a similarity score in $[0, 1]$. The more similar two annotations $g_e^i \in g_e$ and $g_k^j \in g_k$ the higher $s(g_e^i, g_k^j) \in [0, 1]$ and define $s(g_e^i, g_k^j)$ by:

$s(g_e^i, g_k^j)$ is 0, if $(s^i \neq s^j \wedge p^i \neq p^j \wedge o^i \neq o^j)$, $\frac{1}{3}$, if $(s^i = s^j \wedge p^i \neq p^j \wedge o^i \neq o^j)$ or $s^i \neq s^j \wedge p^i = p^j \wedge o^i \neq o^j)$ or $(s^i \neq s^j \wedge p^i \neq p^j \wedge o^i = o^j)$, $\frac{2}{3}$, if $(s^i = s^j \wedge p^i = p^j \wedge o^i \neq o^j)$ or $(s^i \neq s^j \wedge p^i = p^j \wedge o^i = o^j)$ or $(s^i = s^j \wedge p^i \neq p^j \wedge o^i = o^j)$, and 1, if $(s^i = s^j \wedge p^i = p^j \wedge o^i = p^j)$.

Calculating the G-similarity between g_e and g_k requires annotation-wise comparison of each annotation in g_e with all annotations in g_k using the similarity score $s(g_e^i, g_k^j)$ for all i and j. Matrix $\mathcal{M}$ is an $m \times n$ matrix, where m is the number of rows and n is the number of columns. $\mathcal{M}$ represents all possible similarity scores between annotations in g_e and g_k where $m = |g_e|$ and $n = |g_k|$, such that $a_{i,j}$ represents the similarity score for $s(g_e^i, g_k^j)$. It is possible that two annotations within two ADB have nothing in common. Hence, we use $\mathcal{M}$ to identify the best match for each annotation in g_e and all annotations in g_k, and vice versa. $v^c \in \mathbb{R}^n$, with $v_j^c = max_i\ a_{i,j}$ represents the similarity vector containing for each annotation in g_e the highest possible similarity score and $v^r \in \mathbb{R}^m$, with $v_i^r = max_j\ a_{i,j}$ represents the similarity vector containing for each annotation in g_k the highest possible similarity score. The G-similarity is defined as $Sim_G(g_e, g_k) = \frac{1}{2} \cdot (\overline{v^c} + \overline{v^r})$, where $\overline{v^c}$ and $\overline{v^r}$ represents the average value of the similarity vectors taking the ratio between high and low similarity scores into account such that two ADBs g_e and g_k sharing only a small number of high similarity scores and a high number of low similarity scores have a small G-similarity. We normalize $Sim_G(g_e, g_k)$ to the interval $[0, 1]$.

C. Iterative Annotation Enrichment Algorithm

In this section we present the iterative annotation enrichment algorithm in Algorithm 1, which bases on Dempster et al. [21]. Their EM-algorithm estimates the maximum likelihood of parameters handling unobserved variables alternating between the expectation and maximization step. The expectation step creates a function for the expectation of log-likelihood using the present values for the parameters. The maximization step calculates the parameters maximizing the expected log-likelihood in the expectation step. The expectation step of Algorithm 1 identifies d_e-related documents, represented as $\mathcal{D}^{d_e}$, using D- and G-similarity and calculates for all annotations $\mathcal{G}^{d_e}$ the ERV value. The maximization step calculates the new average G-similarity optimizing the ERVs in the next expectation step. We define ERV to estimate only the annotations in $\mathcal{G}^{d_e}$ describing the semantic meaning of the content from document d_e as $ERV_t^{d_e} = \overline{Sim_{D_t}} \cdot \overline{Sim_{G_t}} \cdot f(t)$, where $\overline{Sim_{D_t}}$ is the average D-similarity of documents $d \in \mathcal{D}^{d_e}$ such that g contains annotation t, $\overline{Sim_{G_t}}$ is the average G-similarity of all ADBs $g \in \mathcal{G}^{d_e}$ containing annotation t and $f(t)$ is the frequency of $g \in \mathcal{G}^{d_e}$ containing annotation t. Obviously, the definition for the ERV depends on the annotations we are interested in. We include the frequency to increase the rank of recurrent annotations. The average D-similarity of documents where the corresponding ADBs contain annotation t is given by $\overline{Sim_{D_t^{d_e}}}$. $\overline{Sim_{G_t^{d_e}}}$ represents the average G-similarity containing annotation t and $\overline{ERV^{d_e}}$ is the average ERV of all annotations in d_e. There are two ways leading to a high ERV. First, D- and G-similarity between d_e and $\mathcal{D}^{d_e}$ is high which means the text of each $d \in \mathcal{D}^{d_e}$ and d_e is semantically related. Second, the number of documents in $\mathcal{D}^{d_e}$ containing annotation t is high. Thus, enriching ADB of d_e with annotation t occurring in many other ADB may add value to the ADB of d, because it seems to be generic or very specific for those documents. The input parameters of Algorithm 1 are document d_e, g_e, $\mathcal{D} \setminus \{d_e\}$, and D-similarity selection threshold τ. The output is the optimal ADB g_e'. In the E-Step, the algorithm updates variable erv_t for each annotation t in $\mathcal{G}_e^d$. The algorithm adds annotations with high ERV to ADBs and ignores annotations with low ERV. In the M-Step, the algorithm updates the average G-similarity $\overline{Sim_{G^{d_e}}}$ which is part of the termination condition in line 5.

Algorithm 1 Iterative Annotation Enrichment

1: **Input**: d_e, g_e, $\mathcal{D} \setminus \{d_e\}$, τ
2: **Output**: g_e'
3: **Define:** $\epsilon = 0.1$, $\mathcal{D}^{d_e}$, $\mathcal{D}'^{d_e}$, $\mathcal{G}^{d_e}$, g_e'
4: **Initialize:** $\overline{Sim_{G^{d_e}}} = \epsilon$, $\overline{Sim'_G} = \overline{Sim_{G^{d_e}}} - \epsilon$, $\mathcal{D}^{d_e} = \emptyset$, $\mathcal{G}^{d_e} = \emptyset$, $erv_t^{d_e} = 0$, $g_e' = \emptyset$
5: **while** $|\overline{Sim_{G^{d_e}}} - \overline{Sim'_{G^{d_e}}}| \geq \epsilon$ **and** $\overline{Sim_{G^{d_e}}} > \overline{Sim'_G}$ **do**
6: $g_e' \leftarrow g_e$
7: $\mathcal{D}^{d_e} \leftarrow \emptyset$ ▷ E-Step
8: **for each** $d_k \in \mathcal{D}$ **do**
9: **if** $Sim_D(d_e, d_k) > \tau$ **and** $Sim_G(g_e', g_k) > \overline{Sim_{G^{d_e}}}$ **then**
10: $\mathcal{D}^{d_e} \leftarrow \mathcal{D}^{d_e} \cup \{d_k\}$
11: **for each** $t \in \mathcal{G}^{d_e}$ **do**
12: $erv_t^{d_e} \leftarrow erv_t^{d_e} + ERV_t^{d_e}$
13: **for each** $t \in \mathcal{G}^{d_e}$ **do**
14: **if** $ERV_t^{d_e} > \overline{ERV}^{d_e}$ **then**
15: $g_e' \leftarrow g_e' \cup \{t\}$ ▷ M-Step
16: $\overline{Sim'_{G^{d_e}}} = \overline{Sim_{G^{d_e}}}$
17: $\overline{Sim_{G^{d_e}}} = \frac{\sum_{k=1}^{|\mathcal{D}^{d_e}|} Sim_{G^{d_e}(g_e, g_k)}}{|\mathcal{D}^{d_e}|}$
18: **return** g_e'

IV. Conclusion

In this paper, we have introduced an unsupervised corpus-driven annotation enrichment approach considering the com-

position of documents and have used an EM-like algorithm to enrich weakly annotated documents with meaningful annotations of related documents from the same corpus. To the best of our knowledge this is the first corpus-driven annotation enrichment model rest upon a combination of two holistic measures to enrich documents with annotations of related documents. In the context of our case study, we conclude that the approach enriches ADBs with annotations representing a valuable content description. Algorithm 1 has a positive predictive value of up to 0.72 for different documents in dataset 1 and 0.96 for some documents in the second dataset. In future work we will extend the algorithm to independently handle D- and G-similarity. Actually, the algorithm cannot identify annotations describing the semantics of documents when only D- or G-similarity is high. Another idea is to learn the thresholds for D- and G-Similarity for each corpus individually.

References

[1] Diego Collarana et al. "MINTE: semantically integrating RDF graphs". In: *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*. ACM. 2017, p. 22.

[2] Gabor Angeli, Melvin Johnson Premkumar, and Christopher D Manning. "Leveraging linguistic structure for open domain information extraction". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. 2015.

[3] Ronald Fagin et al. "Document spanners: A formal approach to information extraction". In: *Journal of the ACM (JACM)* 62.2 (2015), p. 12.

[4] Wei Shen, Jianyong Wang, and Jiawei Han. "Entity linking with a knowledge base: Issues, techniques, and solutions". In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (2015), pp. 443–460.

[5] Dong, Xin Luna and Gabrilovich, Evgeniy and Heitz, Geremy and Horn, Wilko and Murphy, Kevin and Sun, Shaohua and Zhang, Wei. "From data fusion to knowledge fusion". In: *Proceedings of the VLDB Endowment* 7.10 (2014), pp. 881–892.

[6] Hamish Cunningham et al. "Getting more out of biomedical documents with GATE's full lifecycle open source text analytics". In: *PLoS computational biology* 9.2 (2013), e1002854.

[7] Aldo Gangemi et al. "Automatic typing of DBpedia entities". In: *International Semantic Web Conference*. Springer. 2012, pp. 65–81.

[8] Valentina Presutti, Francesco Draicchio, and Aldo Gangemi. "Knowledge extraction based on discourse representation theory and linguistic frames". In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer. 2012, pp. 114–129.

[9] Linyuan Lü and Tao Zhou. "Link prediction in complex networks: A survey". In: *Physica A: statistical mechanics and its applications* 390.6 (2011).

[10] Carlson, Andrew and Betteridge, Justin and Kisiel, Bryan and Settles, Burr and Hruschka Jr, Estevam R and Mitchell, Tom M. "Toward an Architecture for Never-Ending Language Learning." In: *AAAI*. Vol. 5. 2010.

[11] Pavlina Fragkou et al. "BOEMIE Ontology-Based Text Annotation Tool." In: *LREC*. Citeseer. 2008.

[12] Thomson Reuters. "OpenCalais". In: *Retrieved June* 16 (2008).

[13] Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. MIT press, 2007.

[14] Suchanek, Fabian M and Kasneci, Gjergji and Weikum, Gerhard. "Yago: a core of semantic knowledge". In: *Proceedings of the 16th international conference on World Wide Web*. ACM. 2007, pp. 697–706.

[15] Alexander Yates et al. "Textrunner: open information extraction on the web". In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics. 2007, pp. 25–26.

[16] Lise Getoor and Christopher P Diehl. "Link mining: a survey". In: *Acm Sigkdd Explorations Newsletter* 7.2 (2005), pp. 3–12.

[17] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003).

[18] Andrew Kachites McCallum. "MALLET: A Machine Learning for Language Toolkit". http://mallet.cs.umass.edu. 2002.

[19] Víctor Manuel Sánchez Valencia. *Studies on natural logic and categorial grammar*. Universiteit van Amsterdam, 1991.

[20] Stuart Geman and Donald Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[21] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38.

[22] Ernst Hellinger. "Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen." In: *Journal für die reine und angewandte Mathematik* 136 (1909), pp. 210–271.

[23] Ce Zhang. "DeepDive: a data management system for automatic knowledge base construction". PhD thesis. The University of Wisconsin-Madison.